

Riemannian Tensor Completion with Side Information

Tengfei Zhou, Hui Qian*, Zebang Shen, Congfu Xu

College of Computer Science and Technology, Zhejiang University, China
{zhoutengfei,qianhui,shenzebang,xucongfufu}@zju.edu.cn

Abstract

Riemannian optimization methods have shown to be both fast and accurate in recovering a large-scale tensor from its incomplete observation. However, in almost all recent Riemannian tensor completion methods, only low rank constraint is considered. Another important fact, side information or features, remains far from exploiting within the Riemannian optimization framework. In this paper, we explicitly incorporate the side information into a Riemannian minimization model. Specifically, a feature-embedded objective is designed to substantially reduce the sample complexity. For such a Riemannian optimization, a particular metric can be constructed based on the curvature of the objective, which leads to a novel Riemannian conjugate gradient descent solver. Numerical experiments suggest that our solver is more efficient than the state-of-the-art when a highly accurate solution is required.

Introduction

Tensor completion, which aims to recover a multidimensional array or tensor from its linear measurements, is ubiquitous in machine learning applications. For example, in video inpainting, one wants to interpolate the whole video based on its partial observations pixels. (Liu et al. 2013; Li et al. 2015); in context-aware recommendation system, one wishes to predicted the unknown ratings based on the historical records (Hidasi and Tikk 2012; Liu, Wu, and Wang 2015); in multilinear and multitask learning, one intends to learn a weight tensor that maps the multidimensional predictors to the responses. (Romera-Paredes et al. 2013; Yu and Liu 2016).

Tensor completion can be encapsulated by a variety of optimization problems. Amongst them, the convex models that formulate the completion task as a tensor nuclear norm penalized regression problem is most popular and well-understood (Wimalawarne, Sugiyama, and Tomioka 2014; Liu et al. 2013; Tomioka, Hayashi, and Kashima 2010). Many research results have proved that the convex model is guaranteed to recover the partial observed tensor under suitable assumptions (Romera-Paredes and Pontil 2013; Yuan and Zhang 2015). However, application of these models to a large-scale instance is difficult, since solving them

involves iterative Singular Value Decomposition (SVD) of multiple huge matrices, which may be computationally prohibitive (Gandy, Recht, and Yamada 2011).

Besides the convex models, tensor completion can also be modeled by nonconvex regression constrained on Riemannian manifolds (Kressner, Steinlechner, and Vandereycken 2014; Kasai and Mishra 2016) that can be solved by the Riemannian optimization framework (Absil, Mahony, and Sepulchre 2009). Empirical comparisons have shown that, in contrast to the nuclear norm solvers, Riemannian solvers use significantly less CPU time to recover the underlying tensor (Kasai and Mishra 2016). This is so because they have much cheaper per-iteration cost as SVD of huge matrices are avoided. Thus, they are more scalable to massive data.

However, to accurately recover the underlying tensor, the Riemannian models demand numerous training samples. Otherwise, the Riemannian models would overfit seriously to the training set, resulting in poor performance over the test set. In matrix completion both theoretical and empirical evidence have shown that exploiting side information is helpful in reducing the sample complexity and improving the accuracy (Xu, Jin, and Zhou 2013; Chiang, Hsieh, and Dhillon 2015). Although matrix completion can be regarded as a special case of tensor completion, directly extending these models to general Riemannian tensor completion is hardly feasible, because they contains non-smooth nuclear norms in the objective functions which can not be solved in the Riemannian optimization framework (Absil, Mahony, and Sepulchre 2009).

In this paper, a novel optimization model for tensor completion is proposed to explicitly incorporate the side information. Then, a novel Riemannian metric is constructed from the second-order derivative of the proposed objective function. Such metric induces an adaptive preconditioner that can accelerate the convergence speed of solvers for the proposed model. Finally, an efficient algorithm for the proposed model is derived based on Riemannian conjugate gradient descent method (Sato and Iwai 2015).

As far as we know, this is the first attempt to integrate the side information in a Riemannian tensor completion model. Experimental results show that our solver outperforms state-of-the-art Riemannian solvers both in accuracy and speed.

*Corresponding author

Notations and Preliminaries

We use bold capital letters to denote a matrix, e.g. \mathbf{X} . The column space of matrix \mathbf{X} is denoted by $\text{span}(\mathbf{X})$. We use Euler script to denote a linear subspace, e.g. \mathcal{A} . A tensor is denoted by bold Euler script, e.g. \mathcal{X} . The (i_1, i_2, i_3) -th element of tensor \mathcal{X} is denoted by $\mathcal{X}_{i_1, i_2, i_3}$. Tensor Frobenius norm is denoted by $\|\cdot\|_F$ where $\|\mathcal{X}\|_F = \sqrt{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} (\mathcal{X}_{i_1, i_2, i_3})^2}$. The mode- n matricization of tensor \mathcal{X} is denoted by $\mathcal{X}_{(n)}$. Note that $\mathcal{X}_{(n)}$ is a matrix obtained by arranging the mode- n fibers of \mathcal{X} so that each of them is a column of $\mathcal{X}_{(n)}$. The mode- n product of tensor \mathcal{X} and matrix \mathbf{A} is denoted by $\mathcal{X} \times_n \mathbf{A}$. Note that the mode- n matricization of the product can be expressed as $(\mathcal{X} \times_n \mathbf{A})_{(n)} = \mathbf{A} \mathcal{X}_{(n)}$.

Distance between Subspaces

The Chordal distance (Ye and Lim 2014) between two subspaces is a function of their principle angles. Specifically, let \mathcal{A} and \mathcal{B} be subspaces of \mathbb{R}^n with dimensions d_1 and d_2 respectively. Without loss of generality, suppose $d_1 \leq d_2$. Let $\mathbf{P} \in \mathbb{R}^{n \times d_1}$ and $\mathbf{Q} \in \mathbb{R}^{n \times d_2}$ be their orthogonal bases. The principal angles between \mathcal{A} and \mathcal{B} form a vector $\theta = (\theta_1, \theta_2, \dots, \theta_{d_1})$, such that $\cos(\theta_i)$ is the i -th largest singular value of matrix $\mathbf{P}^T \mathbf{Q}$ (Golub and van Loan 1996). The Chordal distance between subspace \mathcal{A} and \mathcal{B} can be defined by $\text{Chordal}(\mathcal{A}, \mathcal{B}) = \sqrt{d_2 - d_1 + \sum_{i=1}^{d_1} \sin^2(\theta_i)}$. If the dimensions of subspace \mathcal{A} and \mathcal{B} are fixed, the distance between them are usually measured by the following truncated Chordal distance

$$\text{dist}(\mathcal{A}, \mathcal{B}) = \sqrt{\sum_{1 \leq i \leq d_1} \sin^2(\theta_i)}. \quad (1)$$

Obviously $\text{dist}^2(\mathcal{A}, \mathcal{B}) = \text{Chordal}^2(\mathcal{A}, \mathcal{B}) + d_1 - d_2$. And $\text{dist}(\mathcal{A}, \mathcal{B}) = 0$ if and only if $\mathcal{A} \subset \mathcal{B}$; $\text{dist}(\mathcal{A}, \mathcal{B}) = \sqrt{d_2}$ if and only if $\mathcal{A} \perp \mathcal{B}$.

Fix Multilinear Rank Manifold

The fix multilinear rank tensor manifold¹ (Kasai and Mishra 2016) is defined as

$$\mathcal{M}_r = \{\text{rank}(\mathcal{X}) = r \mid \mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, r = (r_1, r_2, r_3)\}$$

where the multilinear rank $\text{rank}(\mathcal{X})$ is a vector containing the matrix ranks of its matricizations, that is $\text{rank}(\mathcal{X}) := (\text{rank}(\mathcal{X}_{(1)}), \text{rank}(\mathcal{X}_{(2)}), \text{rank}(\mathcal{X}_{(3)}))$. By tucker decomposition (Kolda and Bader 2009), any tensor $\mathcal{X} \in \mathcal{M}_r$ can be factorized as: $\mathcal{X} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$ where the tuple $(\mathcal{G}, \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3)$ belongs to the following product manifold

$$\mathbb{R}_*^{r_1 \times r_2 \times r_3} \times \text{St}(r_1, n_1) \times \text{St}(r_2, n_2) \times \text{St}(r_3, n_3) \triangleq \overline{\mathcal{M}}_r$$

in which $\mathbb{R}_*^{r_1 \times r_2 \times r_3}$ is the open manifold of tensors whose matricizations are all of full rank; $\text{St}(r, n)$ is the Stiefel manifold of $n \times r$ matrices with orthogonal columns (Wen and

Yin 2013). Since the tuple $(\mathcal{G}, \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3)$ provides a representation of \mathcal{X} , we call it a Tucker representation of \mathcal{X} . There is a natural submersion $\pi : \overline{\mathcal{M}}_r \rightarrow \mathcal{M}_r$ between the two manifolds where

$$\pi(\mathcal{G}, \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3) = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3.$$

The submersion induces an equivalent relationship \sim on $\overline{\mathcal{M}}_r$ where $(\mathcal{G}, \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3) \sim (\mathcal{G}', \mathbf{U}'_1, \mathbf{U}'_2, \mathbf{U}'_3)$ if and only if $\pi(\mathcal{G}, \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3) = \pi(\mathcal{G}', \mathbf{U}'_1, \mathbf{U}'_2, \mathbf{U}'_3)$, which means that two elements of the product manifold $\overline{\mathcal{M}}_r$ are equivalent when they represent the same tensor. Because $\pi(\cdot)$ is a submersion from $\overline{\mathcal{M}}_r$ to \mathcal{M}_r , by Proposition 3.3.3 of (Abraham, Marsden, and Ratiu 2012), the equivalent relationship is regular and \mathcal{M}_r is isomorphic to the quotient manifold $\overline{\mathcal{M}}_r / \sim$ namely

$$\mathcal{M}_r \simeq \overline{\mathcal{M}}_r / \sim.$$

Thus, we can say that \mathcal{M}_r is a quotient manifold of $\overline{\mathcal{M}}_r$, and $\overline{\mathcal{M}}_r$ is the total space of \mathcal{M}_r .

For simplicity we denote the elements of $\overline{\mathcal{M}}_r$ by $\overline{\mathcal{X}}$. We also use $\overline{\mathcal{X}}$ to denote a Tucker representation of tensor \mathbf{X} to lighten the notation. The meaning is clear from the context. Suppose $\overline{\mathcal{X}} = (\mathcal{G}, \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3)$. The tangent space of $\overline{\mathcal{M}}_r$ at $\overline{\mathcal{X}}$ is:

$$T_{\overline{\mathcal{X}}} \overline{\mathcal{M}}_r = \mathbb{R}^{r_1 \times r_2 \times r_3} \times T_{\mathbf{U}_1} \text{St}(r_1, n_1) \times T_{\mathbf{U}_2} \text{St}(r_2, n_2) \times T_{\mathbf{U}_3} \text{St}(r_3, n_3) \quad (2)$$

where $T_{\mathbf{U}} \text{St}(r, n)$ is the tangent space of Stiefel manifold. And the tangent vector $(\eta_{\mathcal{G}}, \eta_1, \eta_2, \eta_3) \in T_{\overline{\mathcal{X}}} \overline{\mathcal{M}}_r$ is often abbreviated to $\eta_{\overline{\mathcal{X}}}$.

Vanilla Riemannian Tensor Completion

Let $\mathcal{R} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be the underlying tensor aimed to be recovered. Suppose the entries of \mathcal{R} are only known at some index $(i_1, i_2, i_3) \in \Omega$ where Ω is an index set with cardinality $|\Omega|$. Suppose the multilinear rank of \mathcal{R} is equal to $r = (r_1, r_2, r_3)$. The existing Riemannian tensor completion model can be expressed as follows.

$$\min_{\mathcal{X}} \frac{1}{2} \|\mathcal{P}_{\Omega}(\mathcal{X} - \mathcal{R})\|_F^2 \quad \text{s.t. } \mathcal{X} \in \mathcal{M}_r, \quad (3)$$

where $\mathcal{P}_{\Omega} : \mathbb{R}^{n_1 \times n_2 \times n_3} \rightarrow \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a linear operator such that

$$\mathcal{P}_{\Omega}(\mathcal{X})_{i_1, i_2, i_3} = \begin{cases} \mathcal{X}_{i_1, i_2, i_3} & \text{if } (i_1, i_2, i_3) \in \Omega \\ 0 & \text{otherwise.} \end{cases}$$

Empirical evidence has exhibited that the above model can recover the unknown tensor if the sample size $|\Omega|$ is much larger than the dimension of the search space \mathcal{M}_r (Kasai and Mishra 2016). However, if the observed samples are extremely sparse, the above model may overfit to the observed samples severely, resulting in bad performance on the recovery task. To resolve such issue, we resort to using the side information which has show to be useful in reducing the sample complexity and improving the accuracy of matrix completion.

¹Note that for simplicity we focus on 3-order tensors in this paper, but the developments can be extended to higher order tensors in a straightforward way.

Riemannian Model with Side Information

Following the research works (Chiang, Hsieh, and Dhillon 2015; Xu, Jin, and Zhou 2013), we assume the side information is encoded in the form of feature matrices which describe the latent spaces of the underlying tensor. Such matrices are provided in many tensor completion task, E.G. in context-aware recommendation, we often have accessible to the feature matrices about the user space, the item space and the context space. We also make the assumption that the side information matrices $\mathbf{F}_i \in \mathbb{R}^{n_i \times k_i}$ are of full column rank and $n_i \gg k_i \geq r_i$.

To use these features matrices, we begin with the ideal case where all these feature matrices are perfect in the sense that the latent space is a subset of the feature space: That is,

$$\text{span}(\mathcal{R}_{(i)}) \subset \text{span}(\mathbf{F}_i), 1 \leq i \leq 3. \quad (4)$$

The feature matrices are thought to be perfect because there exist a weight tensor $\mathcal{A} \in \mathbb{R}^{k_1 \times k_2 \times k_3}$ such that the underlying tensor can be expressed as $\mathcal{R} = \mathcal{A} \times_1 \mathbf{F}_1 \times_2 \mathbf{F}_2 \times_3 \mathbf{F}_3$. Recalling that the condition (4) is equivalent to that the Truncated Chordal distance between the two linear subspaces is equal to zero:

$$\text{dist}[\text{span}(\mathbf{F}_i), \text{span}(\mathcal{X}_{(i)})] = 0, 1 \leq i \leq 3. \quad (5)$$

Thus, the side information can be used by adding the above equalities (5) as extra constraints for the optimization problem (3).

In general cases, the feature matrices may not be perfect. We can only expect that the Truncated Chordal distances between the latent spaces and feature spaces are small, that is, $\text{dist}[\text{span}(\mathbf{F}_i), \text{span}(\mathcal{X}_{(i)})] \leq \epsilon_i, 1 \leq i \leq 3$ for some real number $\epsilon_i \geq 0$. To enforce these conditions, we can penalize the truncated Chordal distances and estimate the underlying tensor as follows.

$$\begin{aligned} \min_{\mathcal{X}} f(\mathcal{X}) &:= \frac{1}{2} \|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{R})\|_F^2 + \frac{\lambda|\Omega|}{2} \|\mathcal{X}\|_F^2 \\ &+ \sum_{i=1}^3 \frac{\alpha_i |\Omega|}{2} \text{dist}^2(\text{span}(\mathcal{X}_{(i)}), \text{span}(\mathbf{F}_i)) \quad (6) \\ \text{s.t. } \mathcal{X} &\in \mathcal{M}_r \end{aligned}$$

where λ and α_i are nonnegative parameters of the regularizers. we show that the proposed cost $f(\cdot)$ is smooth over \mathcal{M}_r . by construct a smooth function $\bar{f}(\cdot)$ over the total space $\overline{\mathcal{M}}_r$ that induces $f(\cdot)$ (See Proposition 3.4.5 of (Absil, Mahony, and Sepulchre 2009)). By inducing, we mean that $f(\mathcal{X}) = \bar{f}(\pi(\overline{\mathcal{X}}))$ for every $\mathcal{X} \in \mathcal{M}_r$ and every tucker presentation $\overline{\mathcal{X}}$ of \mathcal{X} . In Riemannian optimization literature the function $\bar{f}(\cdot)$ is called the lifted cost, it is the representative of the cost in the total space. The smooth lifted cost is constructed as follows.

Proposition 1 Suppose \mathbf{P}_i is a orthogonal basis of the feature space $\text{span}(\mathbf{F}_i)$. Then the objective function $f(\mathcal{X})$ of optimization problem (6) can be induced by the following function of the total space $\overline{\mathcal{M}}_r$

$$\begin{aligned} \bar{f}(\overline{\mathcal{X}}) &:= \frac{1}{2} \|\mathcal{P}_\Omega(\pi(\overline{\mathcal{X}}) - \mathcal{R})\|_F^2 + \frac{\lambda|\Omega|}{2} \|\mathcal{G}\|_F^2 \\ &+ \sum_{i=1}^3 \frac{\alpha_i |\Omega|}{2} \text{trace}(\mathbf{U}_i^T (\mathbf{I}_i - \mathbf{P}_i \mathbf{P}_i^T) \mathbf{U}_i) \quad (7) \end{aligned}$$

where \mathbf{I}_i is the identity matrix of size $r_i \times r_i$.

The Riemannian Solver

The proposed model (6) is an instance of smooth minimization problem constrained on a quotient manifold. Generally speaking, the solvers for such class of problems are implemented in the total space (Absil, Mahony, and Sepulchre 2009). And such implementation can be derived once the total space $\overline{\mathcal{M}}_r$ is endowed with a Riemannian metric such that \mathcal{M}_r admits a Riemannian quotient manifold structure under the metric (Absil, Mahony, and Sepulchre 2009).

Metric Tuning

Traditionally, the Riemannian metric has been selected according to the geometric structure of the search space but ignoring the role of the objective function (Absil, Mahony, and Sepulchre 2009). Nevertheless, Mishra and Sepulchre argue that a selectively designed metric taking the curvature of the cost into consideration can greatly improve the convergence speed in (2016). And for quadratic minimization problem constrained on the fix rank matrix manifold, they generate a family of Riemannian metrics from the Hessian of the cost. Applying this family to the proposed optimization problem (6) is computational costly. To circumvent the issue, we focus on inducing the metric via a simplified lifted cost where Ω contains the full set of the indices. Specifically, the simplified cost $g(\overline{\mathcal{X}})$ is defined by

$$\begin{aligned} g(\overline{\mathcal{X}}) &:= \frac{1}{2} \|\pi(\overline{\mathcal{X}}) - \mathcal{R}\|_F^2 + \frac{\lambda|N|}{2} \|\mathcal{G}\|_F^2 \\ &+ \sum_{i=1}^3 \frac{\alpha_i N}{2} \left[\text{trace}(\mathbf{U}_i^T (\mathbf{I}_i - \mathbf{P}_i \mathbf{P}_i^T) \mathbf{U}_i) \right] \quad (8) \end{aligned}$$

where $N = n_1 n_2 n_3$. Let $\eta_{\overline{\mathcal{X}}}, \xi_{\overline{\mathcal{X}}} \in T_{\overline{\mathcal{X}}} \overline{\mathcal{M}}_r$ be two tangent vectors. We construct the Riemannian metric for the total space as follows.

$$\begin{aligned} \langle \eta_{\overline{\mathcal{X}}}, \xi_{\overline{\mathcal{X}}} \rangle_{\overline{\mathcal{X}}} &= D^2 g(\overline{\mathcal{X}}) [\eta_{\overline{\mathcal{X}}}, \xi_{\overline{\mathcal{X}}}] \\ &= \sum_{i=1}^3 \langle \eta_i, \xi_i \mathcal{G}_{(i)} \mathcal{G}_{(i)}^T \rangle + (1 + N\lambda) \langle \eta_{\mathcal{G}}, \xi_{\mathcal{G}} \rangle \\ &+ \sum_{i=1}^3 N \alpha_i \langle \eta_i, (\mathbf{I}_i - \mathbf{P}_i \mathbf{P}_i^T) \xi_i \rangle \quad (9) \end{aligned}$$

where $D^2 g(\overline{\mathcal{X}})$ is the second-order derivative of $g(\cdot)$ and $\langle \cdot, \cdot \rangle$ is the Euclidean inner product. Note that the above metric does not depend on any entries of the tensor \mathcal{R} .

Proposition 2 The fix multilinear rank manifold \mathcal{M}_r admits a structure of Riemannian quotient manifold, if $\overline{\mathcal{M}}_r$ is endowed with the Riemannian metric defined in (9).

Various geometric objects that are ingredients of the solvers are developed. These objects are closely related to the proposed metric. Their expressions are listed in Tab. 1. And detailed mathematical derivations can be found in supplementary material.

Riemannian Conjugate Gradient Descent

Among various Riemannian optimization methods, Riemannian conjugate gradient descent has both fast convergence speed and low per-iteration cost. In the section, we devise

a solver named Conjugate Gradient with Side Information (CGSI) for the problem (6) based on the Riemannian conjugate gradient method. CGSI proceeds as follows. In the k -th iteration, CGSI picks a search direction $\eta^{(k)}$ from current tangent space $T_{\mathcal{X}^{(k)}}\mathcal{M}_r$ and then updates by retraction $\mathcal{X}^{(k+1)} = R_{\mathcal{X}^{(k)}}(t_k \eta^{(k)})$ where $t_k > 0$ is the step size and $R_{\mathcal{X}^{(k)}}(\cdot)$ is the retraction of manifold \mathcal{M}_r . The search direction is computed by the following recurrence

$$\eta^{(k)} = \begin{cases} -\text{grad } f^{(k)} & \text{if } k = 0 \\ -\text{grad } f^{(k)} + \beta^{(k)} \mathcal{T}_k(\eta^{(k-1)}) & \text{otherwise} \end{cases} \quad (10)$$

where we denote the the Riemannian gradient $\text{grad } f(\mathcal{X}^{(k)})$ by $\text{grad } f^{(k)}$, $\beta^{(k)} > 0$ is computed by Fletcher-Reeves formula (Absil, Mahony, and Sepulchre 2009), and $\mathcal{T}_k(\cdot)$ is the vector transport from tangent space $T_{\mathcal{X}^{k-1}}\mathcal{M}_r$ to tangent space $T_{\mathcal{X}^k}\mathcal{M}_r$. Note that the vector transport is required since $\text{grad } f^{(k)}$ and $\eta^{(k-1)}$ belong to different tangent spaces, it makes no sense to combine them linearly. Since \mathcal{M}_r is a Riemannian quotient manifold, both solution sequence $\{\mathcal{X}^{(k)}\}$ and search direction sequence $\{\eta^{(k)}\}$ are represented in the total space. Specifically, $\mathcal{X}^{(k)}$ is represented by its tucker factors $\overline{\mathcal{X}^{(k)}}$ in $\overline{\mathcal{M}_r}$. And $\eta^{(k)}$ is represented by its horizontal lift $\overline{\eta^{(k)}}$ in the horizontal space $\mathcal{H}_{\overline{\mathcal{X}^{(k)}}}$. The horizontal lift can be updated by lifting the recurrence (10) to the total space as follows.

$$\overline{\eta^{(k)}} = \begin{cases} -\overline{\text{grad } f^{(k)}} & \text{if } k = 0 \\ -\overline{\text{grad } f^{(k)}} + \beta^{(k)} \overline{\mathcal{T}_k(\eta^{(k-1)})} & \text{otherwise.} \end{cases} \quad (11)$$

By the theory of Riemannian manifold (Absil, Mahony, and Sepulchre 2009), the horizontal lift of the Riemannian gradient $\text{grad } f^{(k)}$ can be computed by

$$\text{grad } f^{(k)} = \text{grad } \bar{f}(\overline{\mathcal{X}^{(k)}})$$

and the formula for $\text{grad } \bar{f}(\overline{\mathcal{X}^{(k)}})$ is listed in Tab. 1. The horizontal lift of the vector transport $\overline{\mathcal{T}_k(\eta^{(k-1)})}$ can be calculated by the following equation (Absil, Mahony, and Sepulchre 2009)

$$\overline{\mathcal{T}_k(\eta^{(k-1)})} = \Pi_{\overline{\mathcal{X}^{(k)}}}[\Psi_{\overline{\mathcal{X}^{(k)}}}(\overline{\eta^{(k-1)}})]$$

where the expressions of the two projection $\Pi_{\overline{\mathcal{X}^{(k)}}}(\cdot)$ and $\Psi_{\overline{\mathcal{X}^{(k)}}}(\cdot)$ can be found in Tab. 1. We summarize CGSI in Alg. 1. Note that because CGSI only stores elements and tangent vectors of the total space, its memory complexity is $O(\sum_{i=1}^3 n_i k_i + |\Omega|)$. And its per-iteration time complexity is $O(|\Omega| r_1 r_2 r_3 + \sum_{i=1}^3 n_i k_i^2)$. The convergence of CGSI can be directly obtained from the analysis of (Sato and Iwai 2015) and (Ring and Wirth 2012).

Experiments

We validate the performance of the proposed solver on both synthetic and real tensor completion tasks. Four state-of-the-art tensor completion solvers are compared with the proposed one. Three of the baselines are solvers for

Item	Formulation
Horizontal Space $\mathcal{H}_{\overline{\mathcal{X}}}$	$\{\overline{\eta}_{\overline{\mathcal{X}}} \in T_{\overline{\mathcal{X}}} \overline{\mathcal{M}_r} \mathbf{V}_i^\top \eta_i \mathbf{G}_i + \mathbf{W}_i^\top \eta_i \mathbf{G}_{\alpha_i} \text{ is symmetric}\}$
projection of an ambient vector $(\mathbf{Z}_g, \mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3)$ onto $T_{\overline{\mathcal{X}}} \overline{\mathcal{M}_r}$ $\Psi_{\overline{\mathcal{X}}}(\mathbf{Z}_g, \mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3)$	$(\mathbf{Z}_g, \mathbf{Z}_1 - \mathbf{V}_1 \mathbf{S}_1 \mathbf{G}_1^{-1} - \mathbf{W}_1 \mathbf{S}_1 \mathbf{G}_{\alpha_1}^{-1},$ $\mathbf{Z}_2 - \mathbf{V}_2 \mathbf{S}_2 \mathbf{G}_2^{-1} - \mathbf{W}_2 \mathbf{S}_2 \mathbf{G}_{\alpha_2}^{-1},$ $\mathbf{Z}_3 - \mathbf{V}_3 \mathbf{S}_3 \mathbf{G}_3^{-1} - \mathbf{W}_3 \mathbf{S}_3 \mathbf{G}_{\alpha_3}^{-1})$ where \mathbf{S}_i is the solution of : $\begin{cases} \text{sym}(\mathbf{V}_i^T \mathbf{V}_i \mathbf{S}_i \mathbf{G}_i^{-1}) \\ + \text{sym}(\mathbf{W}_i^T \mathbf{W}_i \mathbf{S}_i \mathbf{G}_{\alpha_i}^{-1}) \\ = \text{sym}(\mathbf{U}_i^\top \mathbf{Z}_i). \\ \mathbf{S}_i = \mathbf{S}_i^\top \end{cases}$
Projection of a tangent vector $\eta_{\overline{\mathcal{X}}}$ of total space onto $\mathcal{H}_{\overline{\mathcal{X}}}$ $\Pi_{\overline{\mathcal{X}}}(\eta_{\overline{\mathcal{X}}})$	$(\eta_g + \sum_{1 \leq i \leq 3} \mathcal{G} \times_i \Omega_i, \eta_1 - \mathbf{U}_1 \Omega_1,$ $\eta_2 - \mathbf{U}_2 \Omega_2, \eta_3 - \mathbf{U}_3 \Omega_3)$ where $(\Omega_1, \Omega_2, \Omega_3)$ is the solution of $\begin{cases} \text{skw}(\mathbf{V}_i^T \mathbf{V}_i \Omega_i \mathbf{G}_i + (1 + N\lambda) \mathbf{G}_i \Omega_i \\ + \mathbf{W}_i^T \mathbf{W}_i \Omega_i \mathbf{G}_{\alpha_i}) \\ - (1 + N\lambda) \mathcal{G}_{(i)} (\mathbf{I}_{j_i} \otimes \Omega_{k_i}) \mathcal{G}_{(i)}^\top \\ - (1 + N\lambda) \mathcal{G}_{(i)} (\Omega_{j_i} \otimes \mathbf{I}_{k_i}) \mathcal{G}_{(i)}^\top \\ = \text{skw}(\mathbf{V}_i^T \eta_i \mathbf{G}_i + \mathbf{W}_i^T \eta_i \mathbf{G}_{\alpha_i}) \\ + (1 + N\lambda) \text{skw}(\mathcal{G}_{(i)} (\eta_g)_{(i)}^\top) \end{cases}$ $\Omega_i^\top = -\Omega_i \forall i \in \{1, 2, 3\}$
Retraction of $\overline{\mathcal{M}_r}$ $\mathbf{R}_{\overline{\mathcal{X}}}(\eta_{\overline{\mathcal{X}}})$	$(\mathcal{G} + \eta_g, \text{uf}(\mathbf{U}_1 + \eta_1),$ $\text{uf}(\mathbf{U}_2 + \eta_2), \text{uf}(\mathbf{U}_3 + \eta_3))$
Euclidean gradient of $\bar{f}(\cdot)$ $\nabla \bar{f}(\overline{\mathcal{X}})$	$\nabla_{\mathcal{G}} \bar{f}(\overline{\mathcal{X}}) = \mathcal{S} \times_{i=1}^3 \mathbf{U}_i^\top + N\lambda \mathcal{G}$ $\nabla_{\mathbf{U}_i} \bar{f}(\overline{\mathcal{X}}) = \mathcal{S}_{(i)} (\mathbf{U}_{j_i} \otimes \mathbf{U}_{k_i}) \mathcal{G}_{(i)}$ $+ N\alpha_i \mathbf{W}_i, 1 \leq i \leq 3$ where $\mathcal{S} = \mathcal{P}_\Omega(\pi(\overline{\mathcal{X}}) - \mathcal{R})$
Scaled gradient $\tilde{\nabla} \bar{f}(\overline{\mathcal{X}})$	$\tilde{\nabla}_{\mathcal{G}} \bar{f}(\overline{\mathcal{X}}) = \nabla_{\mathcal{G}} \bar{f}(\overline{\mathcal{X}}) / (1 + N\lambda)$ $\tilde{\nabla}_{\mathbf{U}_i} \bar{f}(\overline{\mathcal{X}}) = \mathbf{E}_i \mathbf{G}_i^{-1} + \mathbf{F}_i \mathbf{G}_{\alpha_i}^{-1}$ where $\mathbf{E}_i = \mathbf{P}_i \mathbf{P}_i^\top \nabla_{\mathbf{U}_i} \bar{f}(\overline{\mathcal{X}})$ and $\mathbf{F}_i = \nabla_{\mathbf{U}_i} \bar{f}(\overline{\mathcal{X}}) - \mathbf{E}_i$
Riemannian gradient of $\bar{f}(\cdot)$ $\text{grad } \bar{f}(\overline{\mathcal{X}})$	$\Psi_{\overline{\mathcal{X}}}(\tilde{\nabla} \bar{f}(\overline{\mathcal{X}}))$

Table 1: Expressions of optimization related objects. The definitions of these objects can be found in (Absil, Mahony, and Sepulchre 2009). In this table we define the following matrices: $\mathbf{V}_i := \mathbf{P}_i \mathbf{P}_i^\top \mathbf{U}_i$, $\mathbf{W}_i := \mathbf{U}_i - \mathbf{V}_i$, $\mathbf{G}_i := \mathcal{G}_{(i)} \mathcal{G}_{(i)}^\top$, $\mathbf{G}_{\alpha_i} = N\alpha_i \mathbf{I}_i + \mathcal{G}_{(i)} \mathcal{G}_{(i)}^\top$ where \mathbf{I}_i is an identity matrix of size $r_i \times r_i$. The indices j_i and k_i mean that $j_i = \max\{k | k \in \{1, 2, 3\}, k \neq i\}$ and $k_i = \min\{k | k \in \{1, 2, 3\}, k \neq i\}$. And the operator $\text{sym}(\cdot)$ and $\text{skw}(\cdot)$ extract the symmetric and skew components of a matrix respectively, i.e. $\text{sym}(\mathbf{A}) = (\mathbf{A} + \mathbf{A}^\top)/2$, $\text{skw}(\mathbf{A}) = (\mathbf{A} - \mathbf{A}^\top)/2$. The operator $\text{uf}(\cdot)$ extracts the orthogonal component of a matrix, that is $\text{uf}(\mathbf{A}) = \mathbf{A}(\mathbf{A}\mathbf{A}^\top)^{-1}$. Note that the linear systems for computing the projectors of tangent space and horizontal space can be solved by conjugate gradient descent with time complexity $O(\sum_{1 \leq i \leq 3} (n_i k_i^2 + T r_i^3))$ where T is the step of CG iterations.

Algorithm 1 CGSI: a Riemannian Conjugate Gradient Descent

Input: Initializer $\overline{\mathcal{X}}^{(0)}$ and tolerance ϵ

- 1: $k = 0$
- 2: **repeat**
- 3: compute current Riemannian gradient $\text{grad } \bar{f}^{(k)}$
- 4: compute the search direction $\overline{\eta}^{(k)}$ by recurrence (11)
- 5: **if** $\langle \overline{\eta}^{(k)}, \text{grad } \bar{f}^{(k)} \rangle_{\overline{\mathcal{X}}^{(k)}} \geq 0$ **then**
- 6: $\overline{\eta}^{(k)} = -\text{grad } \bar{f}^{(k)}$
- 7: **end if**
- 8: choose a step size $t_k \geq 0$
- 9: update by retraction $\overline{\mathcal{X}}^{(k+1)} = R_{\overline{\mathcal{X}}^{(k)}}(t_k \overline{\eta}^{(k)})$
- 10: $k = k + 1$
- 11: **until** $\|\text{grad } \bar{f}^{(k)}\| \leq \epsilon$
- 12: **return** $\overline{\mathcal{X}}^{(k)}$

problem (3). Two of them are recently proposed Riemannian solvers GE (Kressner, Steinlechner, and Vandereycken 2014) and FTC (Kasai and Mishra 2016), and the left one is an Euclidean solver based on Alternate Minimization (AM) (Romera-Paredes et al. 2013). To confirm whether the proposed metric (9) can accelerate the convergence speed of the solver, we also implemented a Riemannian conjugate gradient descent method based on the least square metric developed by (Kasai and Mishra 2016), and we call this baseline FTC with Side Information (FTCSI). The convex solvers such as HalRTC (Liu et al. 2013), Latent (Tomioaka, Hayashi, and Kashima 2010) and Hard (Signoretto et al. 2014) are not compared, since FTC has shown to be state-of-the-art (Kasai and Mishra 2016). All the experiments are performed in Matlab on the same machine with 3.0 GHz Intel E5-2690 CPU and 128GB RAM.

Simulations

In the simulations, we complete a random tensor \mathcal{R} whose size is fixed to $5000 \times 5000 \times 5000$ and multilinear rank to $(10, 10, 10)$. And it is generated by $\mathcal{R} = \mathcal{A} \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \times_3 \mathbf{B}_3$ where $\mathcal{A} \in \mathbb{R}^{10 \times 10 \times 10}$ and $\mathbf{B}_i \in \mathbb{R}^{5000 \times 10}$ are random (multi-dimensional) arrays with i.i.d standard Gaussian entries. The side informations are encoded in three feature matrices. They are generated by $\mathbf{F}_i = \mathbf{B}_i + s \|\mathbf{B}_i\|_F \mathbf{N}_i$ where \mathbf{N}_i is a noise matrix with entries drew from i.i.d normal distribution. The indices of the observed entries Ω are sampled from the full indices set of the $5000 \times 5000 \times 5000$ tensor uniformly at random. Its cardinality $|\Omega|$ is set to $OS \times D$ where $D = 3 \times (5000 \times 10 - 10^2) + 10^3$ is the dimension of the manifolds of $5000 \times 5000 \times 5000$ tensors with multilinear rank $(10, 10, 10)$ and OS is called the Over-Sampling ratio. We compare the five tensor completion solvers under the following four scenarios. In each run the compared solvers are started with the same initializer generated from random, and stopped when either the norm of the gradient is less than 10^{-4} or the number of iterations is more than 300. And the parameters of CGSI and FTCSI are set to the same values as they solve the same problem.

Case 1: influence of sampling ratio We study the number of observed samples on the performance of the com-

pared solvers. We vary the oversampling ratio in the set $OS \in \{0.1, 1, 5\}$ while fixing the noise scale of the feature matrices to 10^{-5} . Then, run the five solvers on each tasks. For each run, we set $\alpha_i, 1 \leq i \leq 3$ are all set to $10/|\Omega|$ and $\lambda = 0$ for CGSI and FTCSI. The parameters of other baselines are set to the defaults. We report the convergence behavior of the compared solvers in Fig. 1(a-c). Note that in Fig. 1(a) the RMSE curve of FTC coincides with that of GE and in Fig. 1(c) the RMSE curve of FTC coincides with that of FTCSI. From Fig. 1(a) and (b), we can see that only CGSI and FTCSI successfully bring the RMSE down below 10^{-2} when OS is smaller than 1. This shows that when the observed entries are scarce, using the side information in the optimization can make a big difference on the accuracy of tensor completion task. And from Fig. 1(a-c), we can see that CGSI converges to the solution faster than FTCSI. This shows that our proposed metric can indeed accelerate the convergence of Riemannian conjugate gradient descent method.

Case 2: influence of noisy side information To study the affect of noisy feature matrix on the performance of the proposed method. We fix the oversampling ratio to $OS = 1$ and vary the noise scale of the feature c matrix in the set $\{10^{-4}, 10^{-3}, 10^{-2}\}$. For CGSI and FTCSI, their parameters α_i are all set to 1 and λ is set to 0. The convergence behavior of the compared methods are reported in Fig. 1(d-f). From these figures we can see that when converging, the RMSE of CGSI and FTCSI are similar. This is because they solve the same problem. And even the feature matrices are noisy, the RMSE of CGSI and FTCSI are much better than the other baselines. These figures also show that CGSI is much faster than FTCSI, which is attributed to that CGSI is endowed with a better Riemannian metric.

Case 3: influence of non-relevant features We consider the performance of the proposed method, when the provided feature matrices \mathbf{F}_i have much more columns than the correct ones \mathbf{B}_i . The matrices $\mathbf{F}_i \in \mathbb{R}^{5000 \times 10(k+1)}$ is generated by augmenting the correct feature matrices \mathbf{B}_i with $10k$ randomly generated columns. That is, we set $\mathbf{F}_i = [\mathbf{B}_i, \mathbf{G}_i] + 10^{-5} \|\mathbf{B}_i\| \mathbf{E}_i$ where $\mathbf{G}_i \in \mathbb{R}^{5000 \times 10k}$ and $\mathbf{E}_i \in \mathbb{R}^{5000 \times 10(k+1)}$ are random matrices with entries drew from i.i.d standard Gaussian distribution. We fix the oversampling ratio to $OS = 1$, and vary the parameter $k \in \{10, 30, 50\}$. For CGSI and FTCSI, $\alpha_i, 1 \leq i \leq 3$ are set to 0.5 and λ is set to 0. The parameters of other baselines are set to the default. We report the convergence behavior of the compared solvers in Fig. 1 (g-i). From these figures we can see that both CGSI and FTCSI successfully bring the RMSE down around 10^{-5} even when the columns of \mathbf{F}_i are 50 times larger than \mathbf{B}_i . And These figures also shows that the proposed solver CGSI converges much faster than FTCSI, which is attributed to CGSI being endowed with a better Riemannian metric.

Case 4: influence of noisy samples We consider the case where the observed entries are noisy by adding a scaled Gaussian noise $\epsilon \mathcal{P}_\Omega(\mathcal{E})$ to $\mathcal{P}_\Omega(\mathcal{R})$ where \mathcal{E} is a noise tensor with i.i.d standard Gaussian entries. We fix the over-

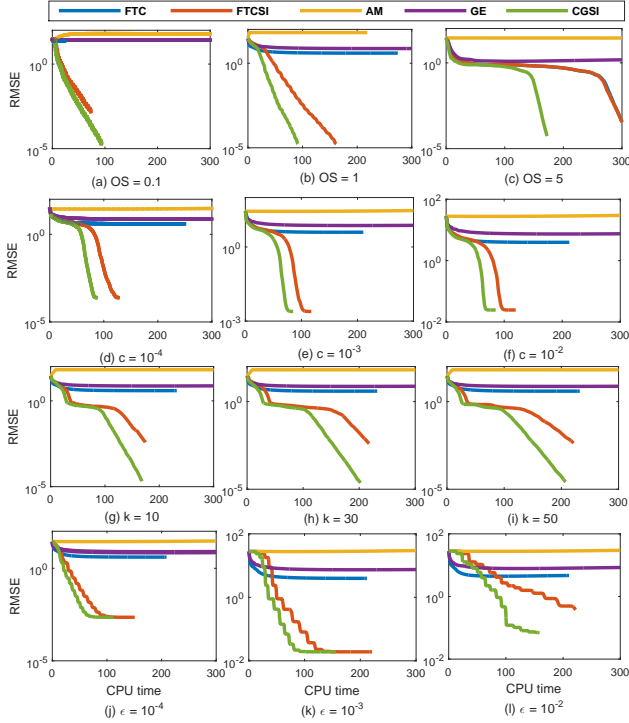


Figure 1: Simulation results of different solvers on the task of tensor completion.

Table 2: Statistics of datasets where ML10M is short for MovieLens10M and ML20M is for MovieLens20M.

	#users	#movies	#slices	#ratings
ML10M	71,567	10,681	731	10,000,054
ML20M	138,493	26,744	731	20,000,263

sampling ratio OS to 1, the noise scale c of feature matrices to 10^{-4} and vary the noise scale of samples such that $\epsilon \in \{10^{-4}, 10^{-3}, 10^{-2}\}$. For CGSI and FTCSI, their parameters are set as follow. When $\alpha_i = 5, 1 \leq i \leq 3$ and $\lambda = 0$. The parameters of other baselines are set to defaults. We report the performance of the compared solvers in Fig. 1 (j-l). From these figures we can see that only the solvers for the proposed model, that is CGSI and FTCSI, bring the RMSE down to the level of noise ϵ when converging. This shows that when the observed entries are few, exploiting the side information can significantly improves the RMSE. Also we can see that CGSI converges much faster than FTCSI, this exhibit that the proposed metric (9) is able to accelerate the convergence of Riemannian conjugate gradient descent method.

Recommender System

In this section, we focus on the specific recommendation task that predicts the rating of an item given by a user at a specific time. Two datasets are considered: MovieLens 10M and MovieLens 20M. The basic statistical information of the two datasets are listed in Tab. 2. For simplicity we

partition the time stamps of both datasets into 731 slices each with the identical time intervals. And the recommendation tasks of the two datasets are transformed into the tensor completion problem of size $71567 \times 10681 \times 731$ and $138493 \times 26744 \times 731$ respectively.

In addition to the rating history, both datasets contain two extra files, one lists the genres of each movie, and the other contains the tags of each movie. Put the genres and tags of the same movie into the same document. We have a corpus containing documents about movies. Then we can extract the feature matrix of movies by applying the text analysis tools. For efficiency, we use Latent Semantic Analysis (LSA) to get the feature matrix which is of size $m \times k$ where m is the number of movies and k is a tuning parameter. Note that as the term-frequency matrix of the corpus are extremely sparse, the feature matrix can be computed via randomized SVD whose time complexity depends linearly in the number of non-zero entries of the term-frequency matrix.

We partition the rating files of both datasets into 80% for training and the remaining 20% for testing. The multilinear rank of all the compared solvers are set to $(10, 10, 1)$. And all the compared methods are initialized by the same randomized value, and are stopped if the training RMSE is below 0.7 or the number of iterations exceeds 100. For both datasets, the parameters of CGSI and FTCSI are set as follows. $\alpha_1 = 0, \alpha_2 = 0.01, \alpha_3 = 0$, and λ is set to $0.001/|\Omega|$. The parameters of other solvers are set to the default. We report the RMSE and CPU time of the compared solvers in Tab. 2. From the table we can see that the solvers for the proposed model, CGSI and FTCSI have the smallest RMSE compared to the other solvers. This shows that our proposed model successfully exploits the side information to improve the accuracy of tensor completion. And the table also shows that the proposed solver CGSI is significantly faster than the compared solvers. This is because the proposed solvers are equipped with an adaptive preconditioner which promotes its convergence speed.

Table 3: Performance of the compared methods on the task of recommendation.

	MovieLens10M		MovieLens20M	
	RMSE	time(s)	RMSE	time(s)
AM	0.9410	441	0.9660	861
GE	0.8212	183	0.8108	430
FTC	0.8187	179	0.8086	418
FTCSI	0.8091	142	0.7985	422
Proposed	0.8055	104	0.7904	284

Conclusion

In this paper, we exploit the side information to improve the accuracy of Riemannian tensor completion. A novel Riemannian model is proposed. To solve the model efficiently, we design a new Riemannian metric. Such metric will induce an adaptive preconditioner for the solvers of the proposed model. Then, we devise a Riemannian conjugate gradient

descent method using the adaptive preconditioner. Empirical results show that our solver outperforms state-of-the-arts.

References

- Abraham, R.; Marsden, J. E.; and Ratiu, T. 2012. *Manifolds, tensor analysis, and applications*, volume 75. Springer Science & Business Media.
- Absil, P.-A.; Mahony, R.; and Sepulchre, R. 2009. *Optimization algorithms on matrix manifolds*. Princeton University Press.
- Chiang, K.-Y.; Hsieh, C.-J.; and Dhillon, I. S. 2015. Matrix completion with noisy side information. In *Advances in Neural Information Processing Systems*, 3447–3455.
- Gandy, S.; Recht, B.; and Yamada, I. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* 27(2):025010.
- Golub, G. H., and van Loan, C. F. 1996. Matrix computations. *Matrix computations/Gene H. Golub, Charles F. Van Loan. Baltimore: Johns Hopkins University Press, 1996.(Johns Hopkins studies in the mathematical sciences)* 1.
- Hidasi, B., and Tikk, D. 2012. Fast als-based tensor factorization for context-aware recommendation from implicit feedback. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 67–82.
- Kasai, H., and Mishra, B. 2016. Low-rank tensor completion: a riemannian manifold preconditioning approach. *arXiv preprint arXiv:1605.08257*.
- Kolda, T. G., and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM review* 51(3):455–500.
- Kressner, D.; Steinlechner, M.; and Vandereycken, B. 2014. Low-rank tensor completion by riemannian optimization. *BIT Numerical Mathematics* 54(2):447–468.
- Li, C.; Zhao, Q.; Li, J.; Cichocki, A.; and Guo, L. 2015. Multi-tensor completion with common structures. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Liu, J.; Musialski, P.; Wonka, P.; and Ye, J. 2013. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(1):208–220.
- Liu, Q.; Wu, S.; and Wang, L. 2015. Cot: Contextual operating tensor for context-aware recommender systems. In *AAAI*, 203–209.
- Mishra, B., and Sepulchre, R. 2016. Riemannian preconditioning. *SIAM Journal on Optimization* 26(1):635–660.
- Ring, W., and Wirth, B. 2012. Optimization methods on riemannian manifolds and their application to shape space. *SIAM Journal on Optimization* 22(2):596–627.
- Romera-Paredes, B., and Pontil, M. 2013. A new convex relaxation for tensor completion. In *Advances in Neural Information Processing Systems*, 2967–2975.
- Romera-Paredes, B.; Aung, H.; Bianchi-Berthouze, N.; and Pontil, M. 2013. Multilinear multitask learning. In *Proceedings of the 30th International Conference on Machine Learning*, 1444–1452.
- Sato, H., and Iwai, T. 2015. A new, globally convergent riemannian conjugate gradient method. *Optimization* 64(4):1011–1031.
- Signoretto, M.; Dinh, Q. T.; De Lathauwer, L.; and Suykens, J. A. 2014. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning* 94(3):303–351.
- Tomioka, R.; Hayashi, K.; and Kashima, H. 2010. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*.
- Wen, Z., and Yin, W. 2013. A feasible method for optimization with orthogonality constraints. *Mathematical Programming* 142(1-2):397–434.
- Wimalawarne, K.; Sugiyama, M.; and Tomioka, R. 2014. Multitask learning meets tensor factorization: task imputation via convex optimization. In *Advances in neural information processing systems*, 2825–2833.
- Xu, M.; Jin, R.; and Zhou, Z.-H. 2013. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in Neural Information Processing Systems*, 2301–2309.
- Ye, K., and Lim, L.-H. 2014. Distance between subspaces of different dimensions. *arXiv preprint arXiv:1407.0900*.
- Yu, R., and Liu, Y. 2016. Learning from multiway data: Simple and efficient tensor regression. In *International Conference on Machine Learning*.
- Yuan, M., and Zhang, C.-H. 2015. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics* 1–38.